GeniE User Manual
Training document
Enhancements to pile/soil modelling and analysis

Table of Contents

GENIE USER MANUAL TRAINING DOCUMENT ENHANCEMENTS TO PILE/SOIL MODELLING AND ANALYSIS 1

TABLE OF CONTENTS 1

1. INTRODUCTION 2

2. STRUCTURE AND LOADS 3

2.1 PILE MODELLING 3
2.2 CREATE THE LOADS 5

3. THE ENVIRONMENT 6

3.1 CREATE THE SOIL TYPES 6
3.2 CREATE THE SCOUR 7
3.3 CREATE THE SOIL CURVES 8
3.3.1 Manual Soil Curves 9
3.3.2 Adjust Soil Curves Data 11
3.3.3 Manual soil curve modelling with unsymmetrical T-Z / Q-Z curves 13
3.4 CREATE THE SOIL DATA 14
3.5 CREATE THE LOCATION 16
3.6 CREATE THE CONDITION 19

4. PILE SOIL ANALYSIS 20

4.1 CREATE THE PILE-SOIL ANALYSIS 20
4.2 EXECUTING THE PILE SOIL ANALYSIS 23

5. THE COMMAND INPUT FILE 24

1. INTRODUCTION

- The main goal of this training example is to describe what is needed to perform a Pile Soil Analysis. The values used, e.g. soil characteristics or layer definitions, are not necessarily realistic. However, the example illustrates how to set up this kind of analysis and many important concepts are explained. Reference is made to the Gensod user documentation for further details.
- A *Pile* in GeniE is considered to be a straight beam, supplied with a set of specific additional properties called Pile Characteristics, which is joined and aligned with an existing beam.
- The usage of piles inside GeniE is restrained to model the interaction of this specific type of beam with the *Soil*. This in turn implies that a natural choice of the coordinate system is to set the xy-plane parallel to the ground or seabed or calm sea-water surface and the z-axis to have an upwards direction. In this sense, the layers/levels are defined only by their z-coordinate. This convention is satisfied in what follows.
- When running GeniE the user is encouraged to move the curser above the dialog light bulbs to get specific information of an input field or of the dialog in general.
- If you want to do the example yourself in GeniE, you may have this document open on your computer so that you can copy and paste the javascript code snippets into the command window in GeniE. The complete command input file is appended at the end of the document.

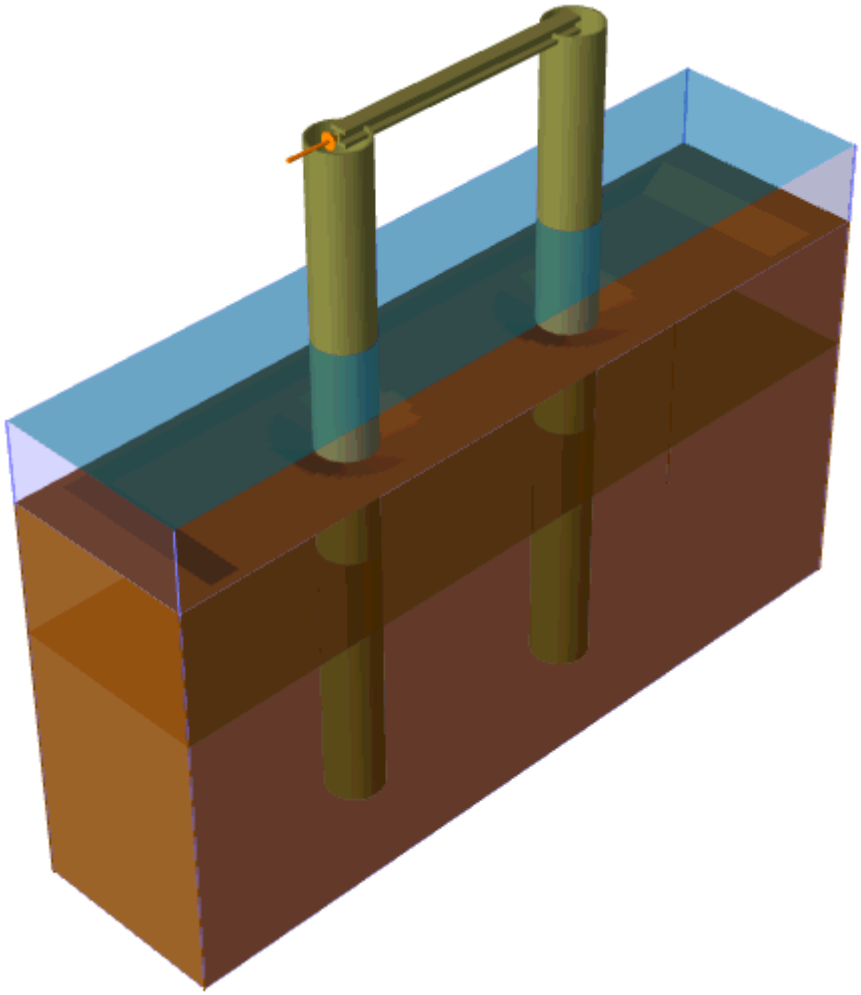


Figure 1.1: A picture of the model used in this example

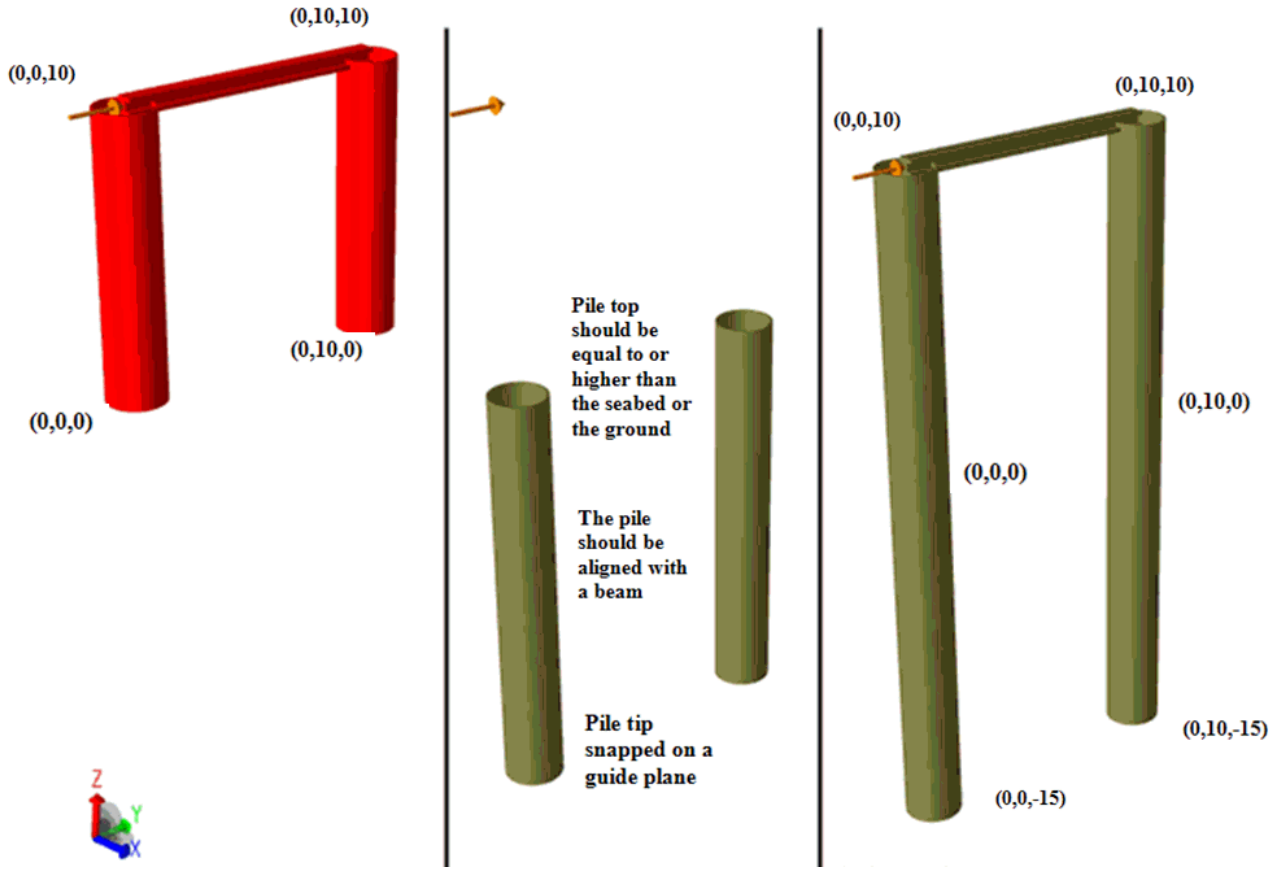
2. STRUCTURE AND LOADS

In this chapter the structure and loads are created. A particular emphasis is put on creating the Pile model.

2.1 Pile Modelling

This shows how to create a simple two-pile structure in GeniE**.**

First we create three beams arranged as in the figure below.



(a) the beam and the support points **(b) the piles** **(c) the complete structure**

Figure 2.1: The structural Pile model

The javascript code is:

```
//Sections and material
Pipe21 = PipeSection(2.1, 0.08);
I600 = ISection(0.6, 0.6, 0.15, 0.15);
Steel = MaterialLinear(255000000, 7850, 2.1e+11, 0.3, 1.2e-5, 0.03);

//Top horizontal beam
Bm1 = Beam(Point(0 m,0 m,10 m), Point(0 m,10 m,10 m));
```

```
Bm1.material = Steel;
Bm1.section = I600;
//Beams connecting piles
Bm2 = Beam(Point(0 m,0 m,10 m), Point(0 m,0 m,0 m));
Bm3 = Beam(Point(0 m,10 m,10 m), Point(0 m,10 m,0 m));
Bm2.material = Steel;
Bm3.material = Steel;
Bm2.section = Pipe21;
Bm3.section = Pipe21;
```

Then, we switch the Straight Beam button into a Pile button by clicking the down arrow next to the button. When the Pile button is pressed you are ready to insert piles (see Figure 2.2).

Figure 2.2: Select Pile in Beam menu

A pile is easily modelled by the following three step procedure (see figure above):

a. Select the pile top (snap to the end beam points)

b. Specify the Z-level of the pile tip in the dialog showing up (do not click anywhere in the model)

c. Align the pile by selecting the alignment beam in the structure.

Since piles are beams, they should have a section and a material assigned to them. The *pile characteristics* allow section stiffness values to be overruled (as compared to ungrouted Pipe section properties) and where special pile tip boundary conditions may be specified. Create the property by right-clicking the Properties > Pile Characteristic folder and selecting New Pile Characteristics (see Figure 2.3). Thereafter assign this property to all piles.

The javascript code is:

```
//Piles
Pile1 = Pile(Point(0 m,0 m,0 m), Point(0 m,0 m,-15 m));
Pile2 = Pile(Point(0 m,10 m,0 m), Point(0 m,10 m,-15 m));
PileType1 = PileCharacteristics(1000 tonne/m^3, tcInfinite);
```

```
Pile2.pileCharacteristics = PileType1;
Pile1.pileCharacteristics = PileType1;
Pile1.section = Pipe21;
Pile2.section = Pipe21;
Pile1.material = Steel;
Pile2.material = Steel;
```

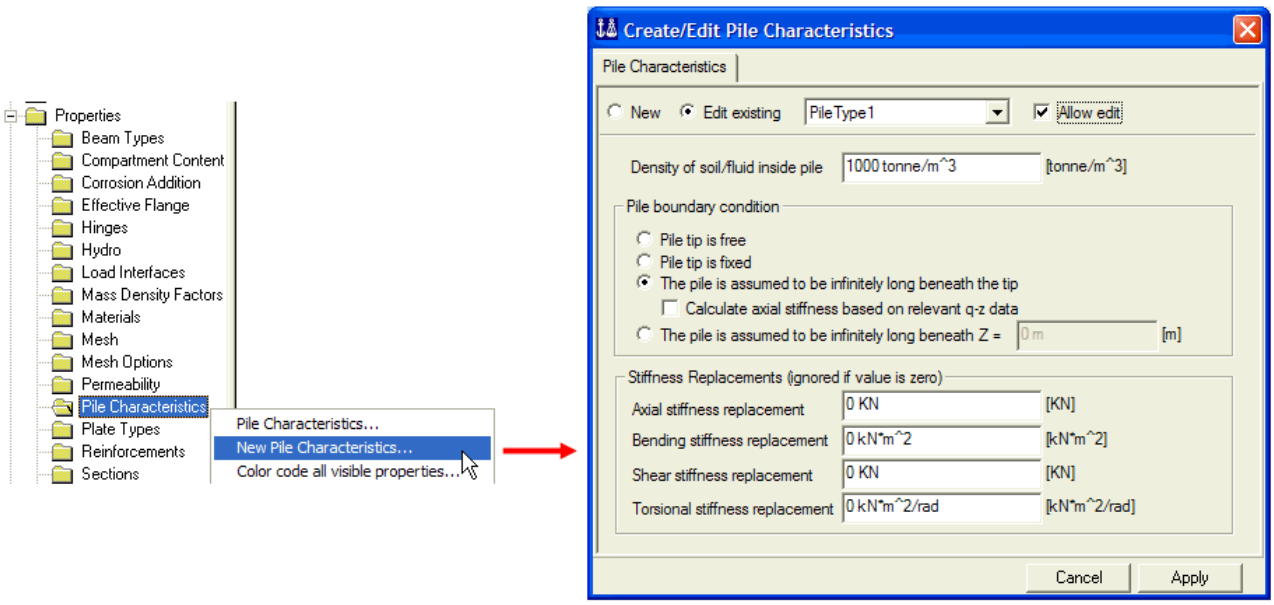


Figure 2.3: Dialog for creating Pile Characteristics

2.2 Create the Loads

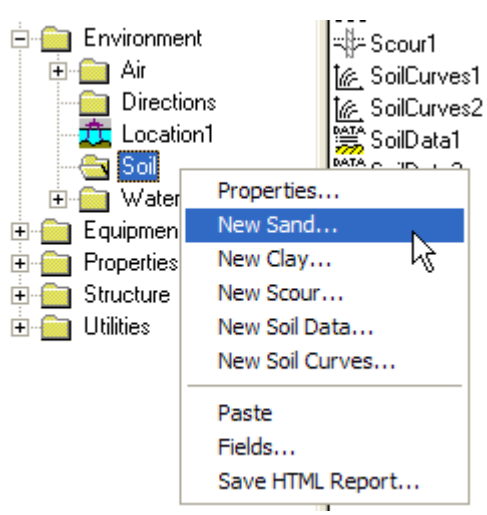
In this example a point load and a gravity load is created. Since this is not specific for a Pile Soil analysis it is not described in further detail.

```
//Loadcase one: sideforce
sideForce = LoadCase();
PLoad1 = PointLoad(sideForce, Point(0 m,0 m,10 m), 0, 100 kN, 0 N, 0 N*m,
0 N*m, 0 N*m);
//Load case two: Gravity
Gravity = LoadCase();
Gravity.setAcceleration(Vector3d(0 m/s^2,0 m/s^2,-9.80665 m/s^2));
Gravity.includeSelfWeight();
Gravity.includeStructureMassWithRotationField();
```

3. THE ENVIRONMENT

In this chapter the Environment model is created. In particular, how to manually define soil curves and modification factors of soil curves is detailed described.

3.1 Create the Soil Types



The soil types are based on the models of Sand and Clay. They can be defined by selecting “New Clay…” and “New Sand…” on the RMB on the browser node “Soil”.

The create/edit soil dialog shows up:

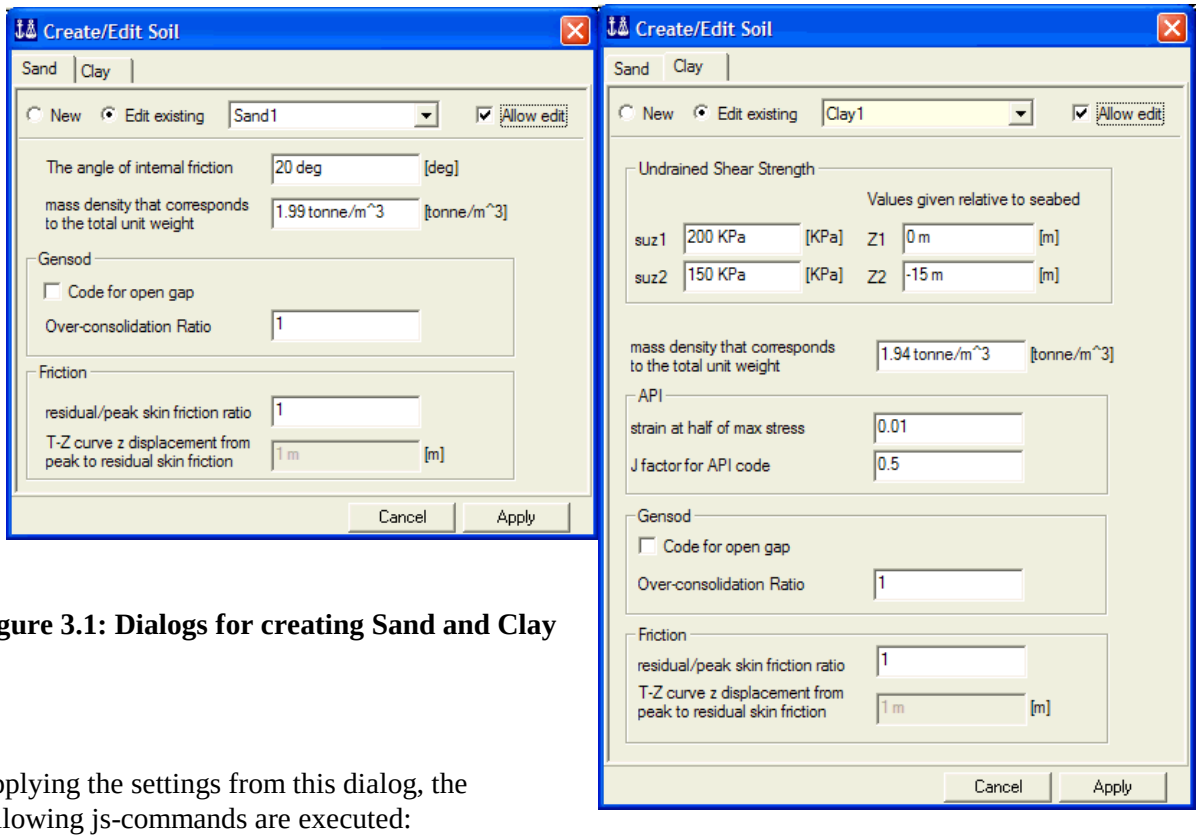


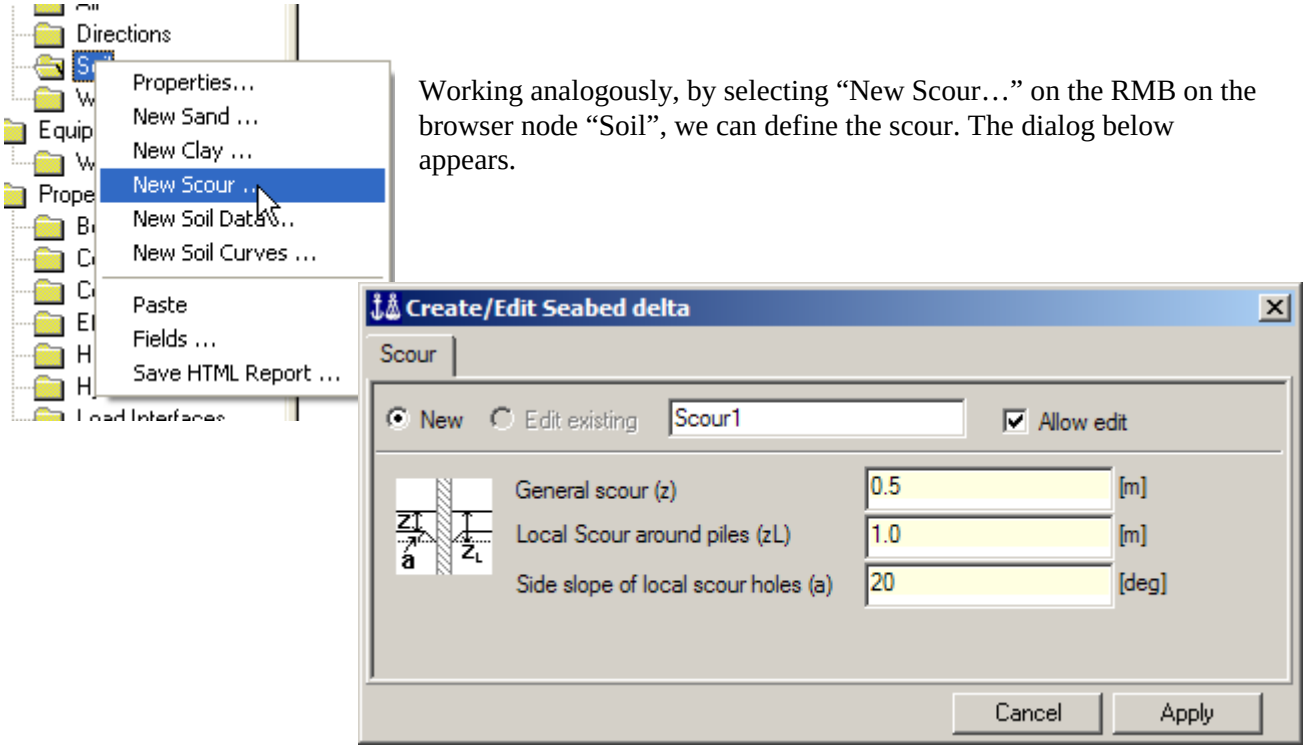
Figure 3.1: Dialogs for creating Sand and Clay

Applying the settings from this dialog, the following js-commands are executed:

```
Sand1 = Sand(false, 1, 1.99 tonne/m^3, 20 deg);
```

```
Clay1 = Clay(false, 1, 1.94 tonne/m^3, 0.01, 200 KPa, 0 m, 150 KPa, -15
m);
Clay1.apiJFactor = 0.5;
```

3.2 Create the Scour



Working analogously, by selecting "New Scour…" on the RMB on the browser node "Soil", we can define the scour. The dialog below appears.

Figure 3.2: Dialog for creating Scour

Applying the settings of the above dialog, the following js-command is executed:

```
Scour1 = Scour(0.5,1.0,20);
```

Note that the seabed level is the top most horizontal line shown in the schematic figure of the dialog (Figure 3.2: Dialog for creating Scour). See also figures on page 17.

3.3 Create the Soil Curves

The Soil Curves are defined in the Soil curves tab of the Soil Properties dialog as shown in the figure below. By selecting “New Soil Curves…” the dialog is activated.

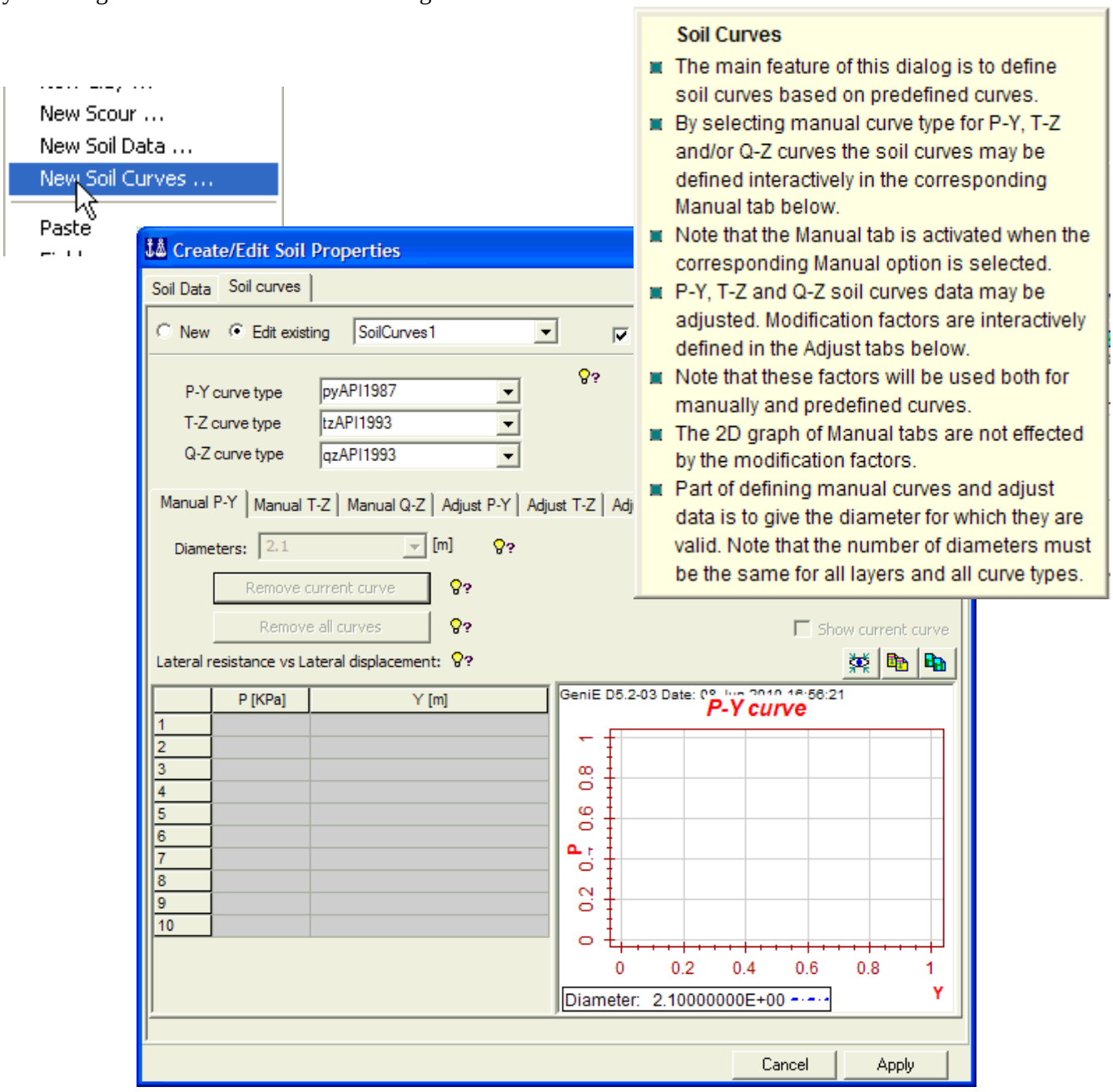

New Scour ...
New Soil Data ...
New Soil Curves ...
Paste
Create/Edit Soil Properties
Soil Data
Soil curves
New
Edit existing
SoilCurves1
P-Y curve type
pyAPI1987
T-Z curve type
tzAPI1993
Q-Z curve type
qzAPI1993
Manual P-Y
Manual T-Z
Manual Q-Z
Adjust P-Y
Adjust T-Z
Diameters:
2.1
[m]
Remove current curve
Remove all curves
Show current curve
Lateral resistance vs Lateral displacement:
P [KPa]
Y [m]
P-Y curve
Diameter: 2.10000000E+00
Cancel
Apply
Soil Curves
The main feature of this dialog is to define soil curves based on predefined curves.
By selecting manual curve type for P-Y, T-Z and/or Q-Z curves the soil curves may be defined interactively in the corresponding Manual tab below.
Note that the Manual tab is activated when the corresponding Manual option is selected.
P-Y, T-Z and Q-Z soil curves data may be adjusted. Modification factors are interactively defined in the Adjust tabs below.
Note that these factors will be used both for manually and predefined curves.
The 2D graph of Manual tabs are not effected by the modification factors.
Part of defining manual curves and adjust data is to give the diameter for which they are valid. Note that the number of diameters must be the same for all layers and all curve types.

Figure 3.3: Dialog for creating soil curves

This is an example of the scripting command creating a soil curve defined only by predefined soil curves:

```
SoilCurves1 = SoilCurves(pyAPI1987,tzAPI1993,qzAPI1993);
```

3.3.1 Manual Soil Curves

GeniE gives the opportunity to define P-Y, T-Z and Q-Z soil curves manually. These curves are defined in the Manual tabs or directly by scripting commands. By selecting the Manual option of the relevant curve type combo box, the corresponding Manual tab is activated. A description of how to define a P-Y manual curve is given in the sequel, in conjunction with the following figure.

Defining P-Y curves

1. Select *pyManual* from the 'P-Y curve type' combo box. The Manual P-Y tab is activated.
2. From the 'Diameters' combo box either select an existing diameter or enter a new value. The combo box is automatically populated with all diameters used by all soil curves of the active *Location* (see page 14).
3. Enter P- and Y-values in the table. Note that the first pair must equal (0,0), all remaining values must be larger than 0 and the Y-values must be in increasing order. Values can be copied from a spreadsheet and pasted into the table.

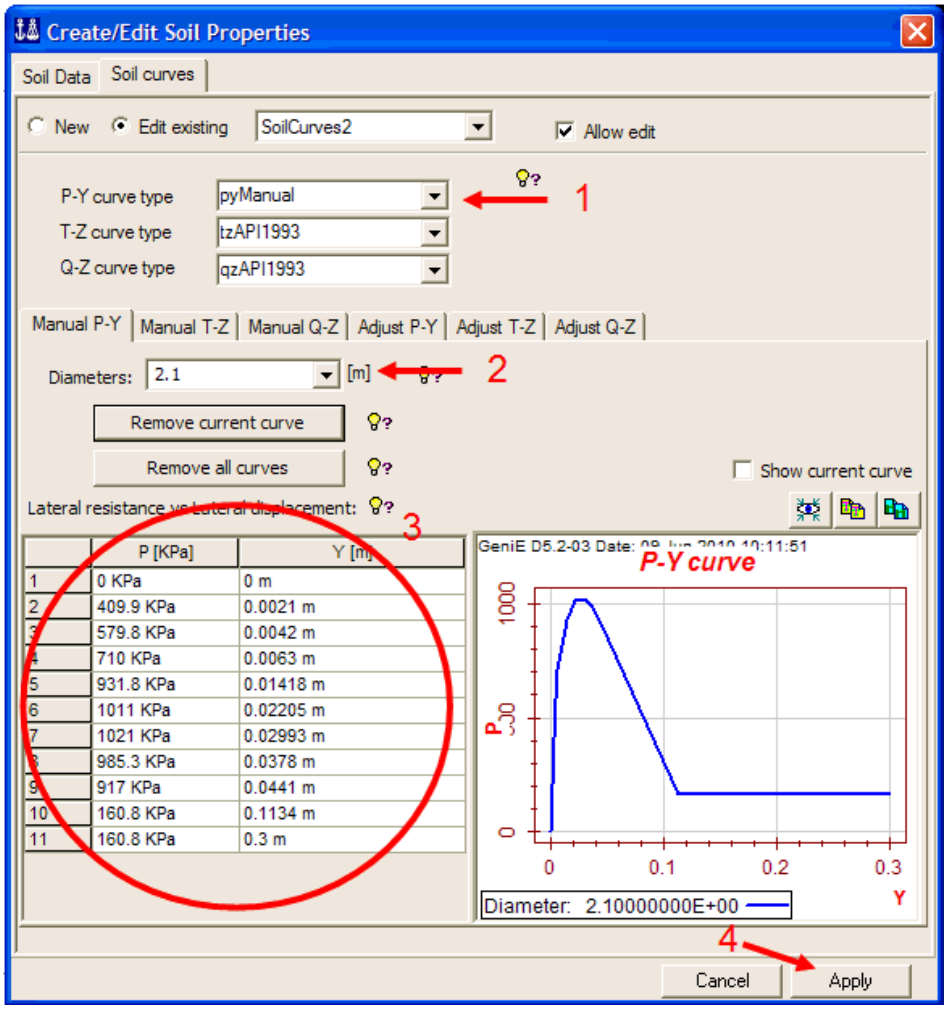


Figure 3.4: The Manual P-Y tab used for defining the manual curve

4. Press 'Apply' when all values are entered. Data will be lost if a new diameter is selected before pressing 'Apply'.

When 'Apply' is pressed in the dialog shown above the following js-commands is executed:

```
SoilCurves2 = SoilCurves(pyManual, tzAPI1993, qzAPI1993);
SoilCurves2.addManualPY(2.1 m, Array(0 m, 0.0021 m, 0.0042 m, 0.0063 m,
0.01418 m, 0.02205 m, 0.02993 m, 0.0378 m, 0.0441 m, 0.1134 m, 0.3 m),
Array(0 KPa, 409.9 KPa, 579.8 KPa, 710 KPa, 931.8 KPa, 1011 KPa, 1021
KPa, 985.3 KPa, 917 KPa, 160.8 KPa, 160.8 KPa));
```

The first parameter is the diameter of the pile, the next is the array of Y-values (lateral displacements) and the last is the array of P-values (lateral resistance).

The 2D graph

In the GUI you get a 2D graphical visualization of the soil curves. The 2D graphs are dynamically updated when values are entered into the table.

By checking the 'Show current curve' check box only the current manual soil curve will be visualized.

Between the 'Show current curve' check box and the graph there are three buttons. From left their roles are:

- Zoom All
- Copy Bitmap
- Copy Metafile

You can get more information by moving the cursor above each button.

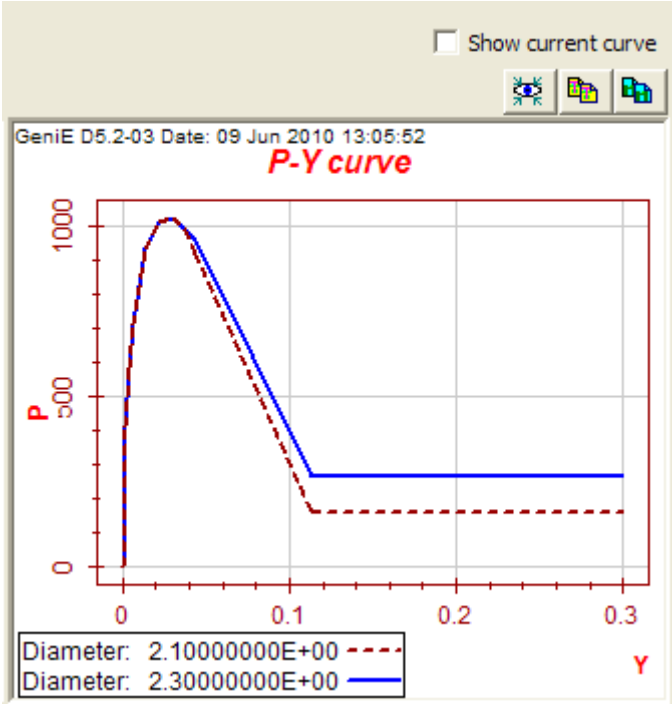
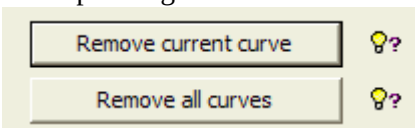


Figure 3.5: The 2D graph

Removing curves

Before pressing the 'Remove current curve'/'Remove all curves' button be sure that you have selected the correct diameter, i.e. have the correct curve in the table. These buttons have two functions. They remove the curve both from the table/graph and from the database. Thus pressing 'Apply' is not needed. If necessary, use the Undo functionality to restore the curves.



When pressing the 'Remove current curve' button the scripting command activated will look like this:

```
SoilCurves2.removeManualPY(2.3);
```

When pressing the 'Remove all curves' button the scripting command activated will look like this:

```
SoilCurves2.removeAllManualPY();
```

3.3.2 Adjust Soil Curves Data

The P-Y, T-Z and Q-Z soil curves may be modified by factors. These factors are defined in the Adjust tabs or directly by scripting commands. Note that these factors will modify all soil curve types, not only those manually defined, by the Gensod program. Moreover, these factors will not influence the 2D graph of the Manual tabs.

Defining factors

The simplest way of defining modification factors is by using the 'Add row' button. For defining P-Y adjust data these are the steps:

1. From the 'Diameters' combo box either select an existing diameter or give a new value. The combo box is automatically populated with all diameters used by all soil curves of the active *Location* (see page 14).
2. Push the 'Add row' button. A row will be appended to the table and filled with default values.

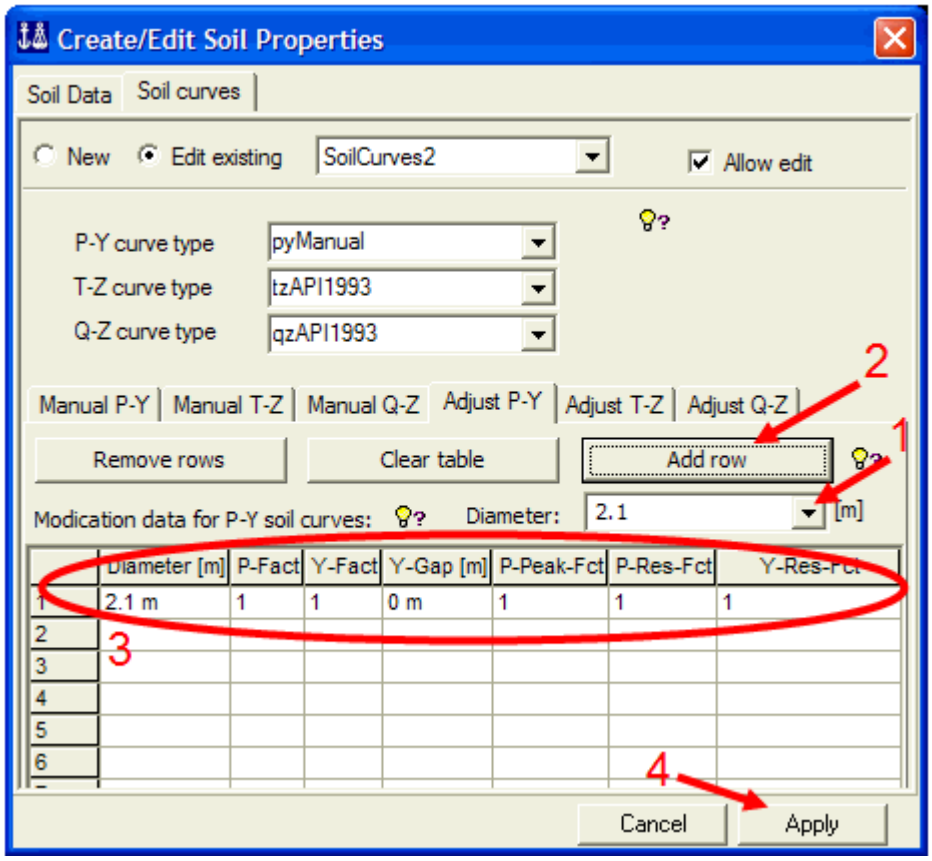


Figure 3.6: The Adjust P-Y tab used for defining the modification factors

3. Modify factors where necessary.
4. Press 'Apply' when all values are entered.

Note that a diameter equal to 0.0 is interpreted to mean that all diameters shall be included.

If a diameter equal to 0.0 is given values for no other diameters shall be defined, i.e. the table shall contain one row, only.

When 'Apply' is pressed for the data in the figure above the scripting commands will look like this:

```
SoilCurves2.addAdjustPY(2.1 m, 1, 1, 0.0 m, 1, 1, 1);
```

Manual definition

As an alternative to using the 'Add row' button the complete row may be manually defined. The Diameter field of each row is a combo box populated with the same diameters as the separate Diameter combo box, plus the value '0.0'. New values cannot be entered into the Diameter cells. Values for all the factors must be entered manually. Alternatively, the values can be copied from a spreadsheet and pasted into the table.

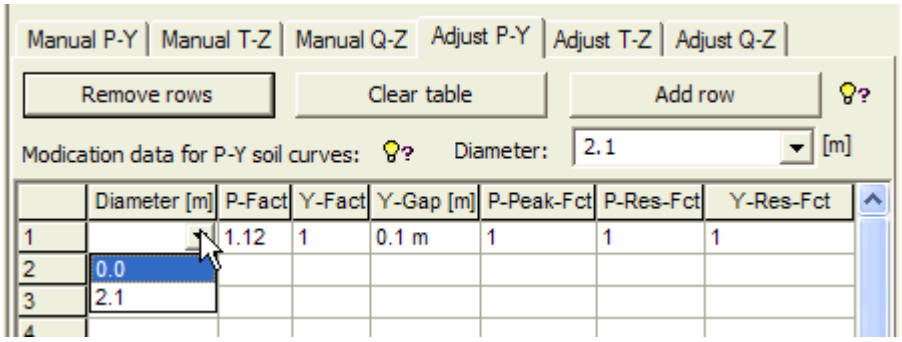


Figure 3.7: Manual definition of modification factors

Remove rows/Clear table

The 'Remove rows'/'Clear table' button in the figure above will remove selected/all rows from the table. To store the changes in the database the 'Apply' button must be pushed.

3.3.3 Manual soil curve modelling with unsymmetrical T-Z / Q-Z curves

GeniE does not allow the user to define the tensile part, i.e. the negative values, of the manually defined T-Z and Q-Z curves. Gensod assumes symmetry when only the curves for positive T-Z or Q-Z values are given, i.e. the tensile part is a mirror image of the compressive part.

The workaround is to use the T-Z/Q-Z modification factors that are set in the 'Adjust T-Z/Q-Z' tabs of the 'Soil Curves' Properties dialog. To modify the tensile resistance use the T-Z or Q-Z modification factor *Tns-Fact*, i.e. the factor that will be multiplied with the resistance of the tensile part of the curve. For example, if 50 % tensile resistance is assumed, set the *Tns-Fac*t equal to 0.5. This is illustrated in the next figure.

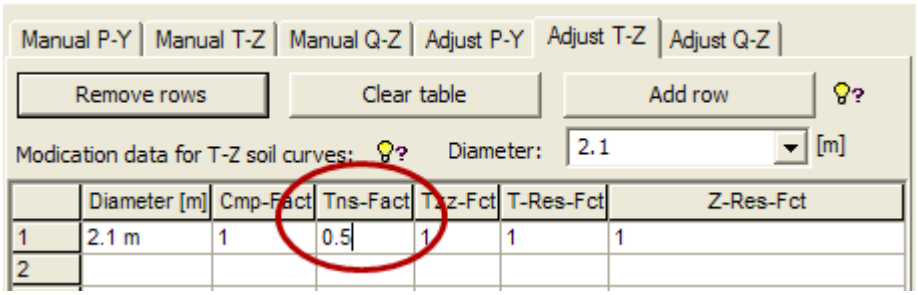


Figure 3.8: Tns-Fact is set equal to 0.5 to achieve 50% tensile resistance

When 'Apply' is pressed for the data in the figure above the scripting commands will look like this:

```
SoilCurves1.addAdjustTZ(2.1 m,1,0.5,1,1,1);
```

How this will influence the T-Z curve is sketched in the example below. Note that the modification of the curve will not be shown in the corresponding 'Manual T-Z/Q-Z' tabs. The modification factors are used by Gensod during the analysis.

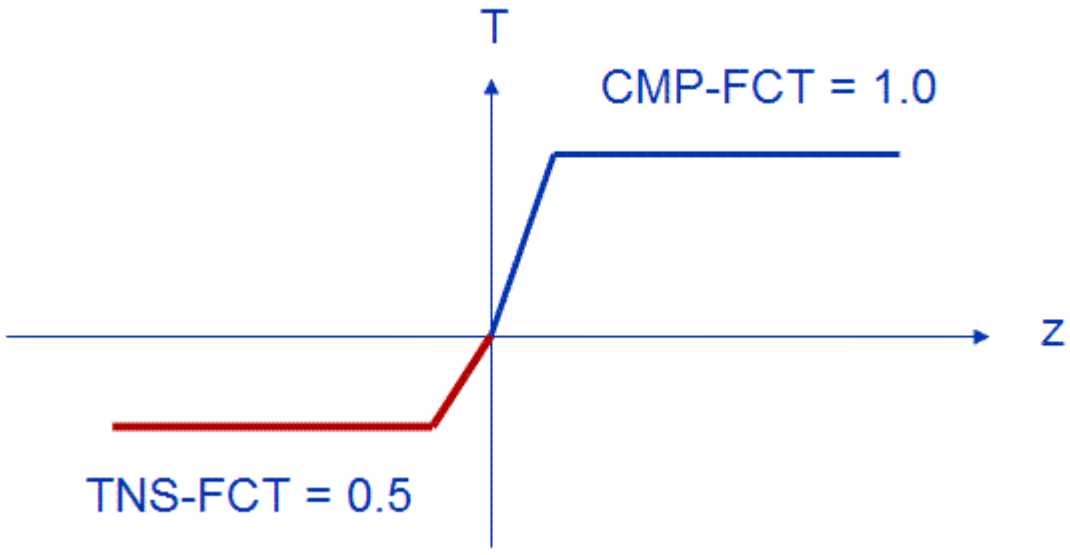
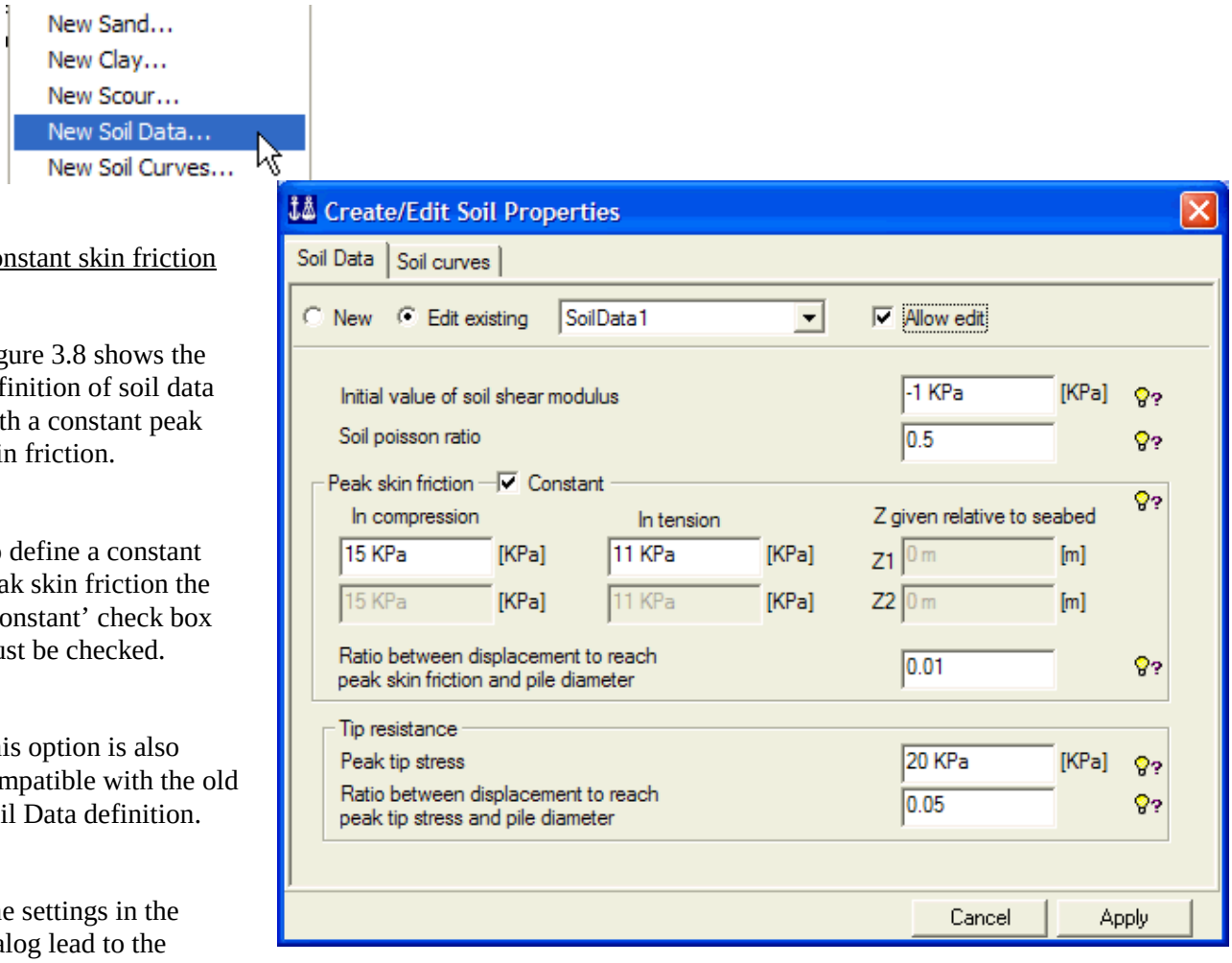


Figure 3.9: Effect of Tns-Fact = 0.5 on the T-Z curve as calculated by Gensod

3.4 Create the Soil Data

The Soil Data are defined in the Soil Data tab of the Soil Properties dialog. By selecting "New Soil Data…" the dialog is activated. The 'Peak skin friction' can either be Constant or Linear Varying with respect to Z. This is described in detail below.



Constant skin friction

Figure 3.8 shows the definition of soil data with a constant peak skin friction.

To define a constant peak skin friction the 'Constant' check box must be checked.

This option is also compatible with the old Soil Data definition.

The settings in the dialog lead to the following js-command:

Figure 3.10: Dialog for creating soil data with constant skin friction

```
SoilData1 = SoilData(-1 KPa, 0.5, 15 KPa, 11 KPa, 0.01, 20 KPa, 0.05);
```

Linearly varying skin friction

Figure 3.9 shows the definition of a linearly varying peak skin friction, which is the default option of the dialog. Note that the Z-values are relative to the seabed and that the orientation of the Z-axis equals that of the global coordinate system in GeniE.

It is assumed that Z1 is greater than Z2.

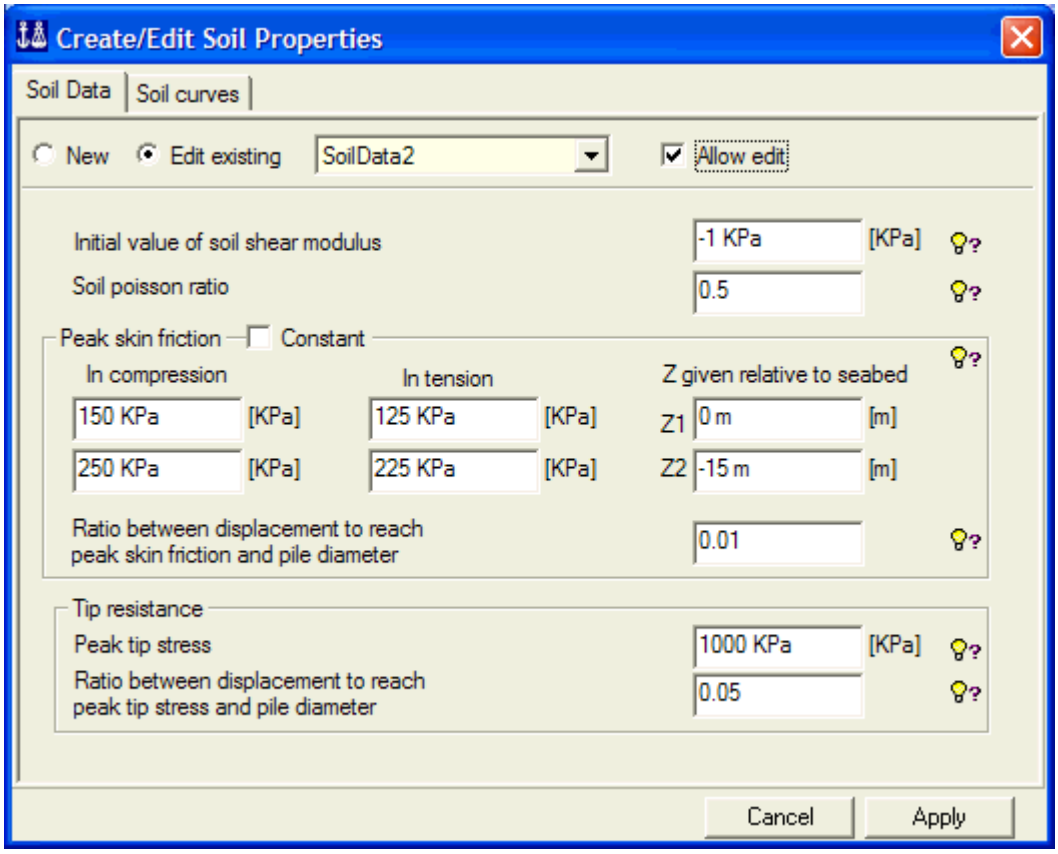


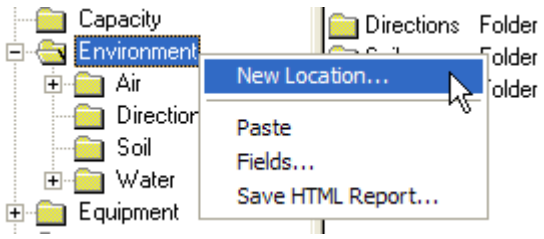
Figure 3.11: Dialog for creating soil data with linear varying skin friction

The above settings lead to the following js-command:

```
SoilData2 = SoilData(-1 KPa, 0.5, 150 KPa, 250 KPa, 125 KPa, 225 KPa, 0
m, -15 m, 0.01, 1000 KPa, 0.05);
```

3.5 Create the Location

Select the “New Location…” entry on the RMB menu, over the browser node “Environment”.



The dialog below shows up. Here you can define the parameters for the *Air*, the *Water* and the *Soil* of the “Environment”.

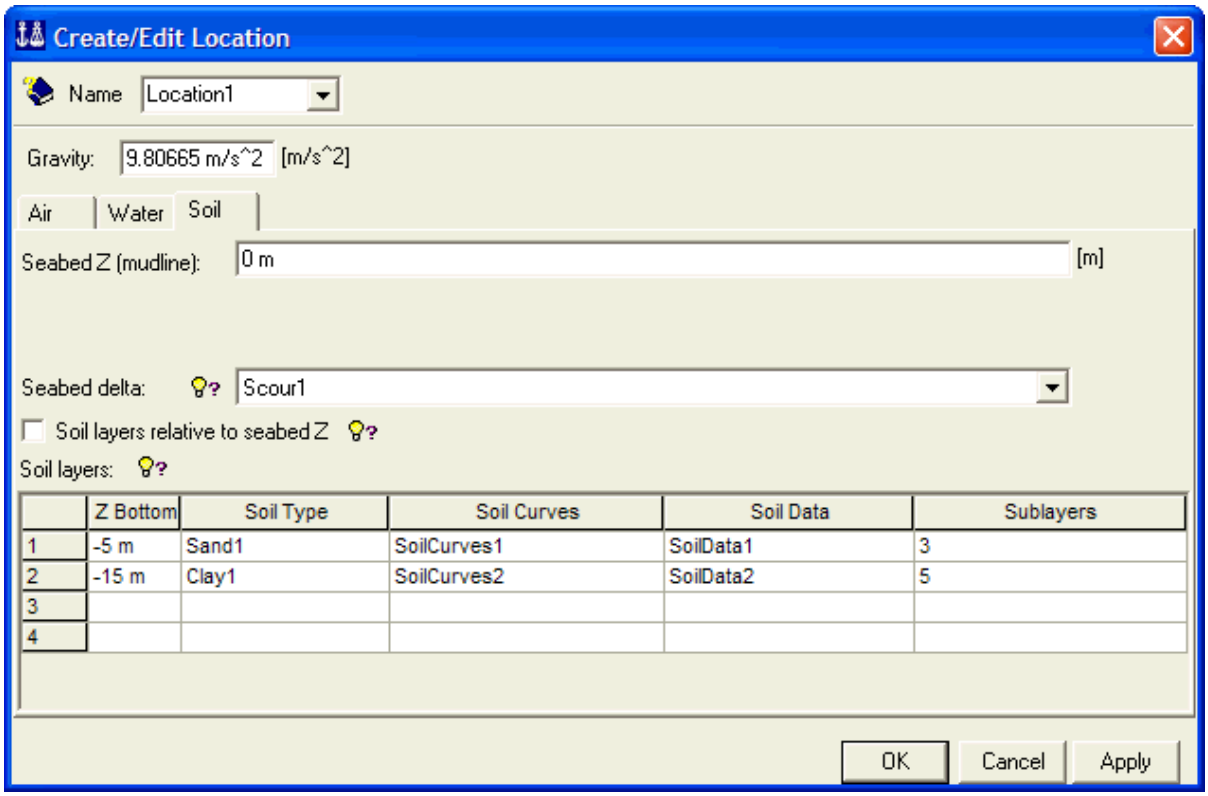

Create/Edit Location
Name Location1
Gravity: 9.80665 m/s^2 [m/s^2]
Air Water Soil
Seabed Z (mudline): 0 m [m]
Seabed delta: Scour1
Soil layers relative to seabed Z
Soil layers:
Z Bottom Soil Type Soil Curves Soil Data Sublayers
1 -5 m Sand1 SoilCurves1 SoilData1 3
2 -15 m Clay1 SoilCurves2 SoilData2 5
3
4
OK Cancel Apply

Figure 3.12: Dialog for creating the Location

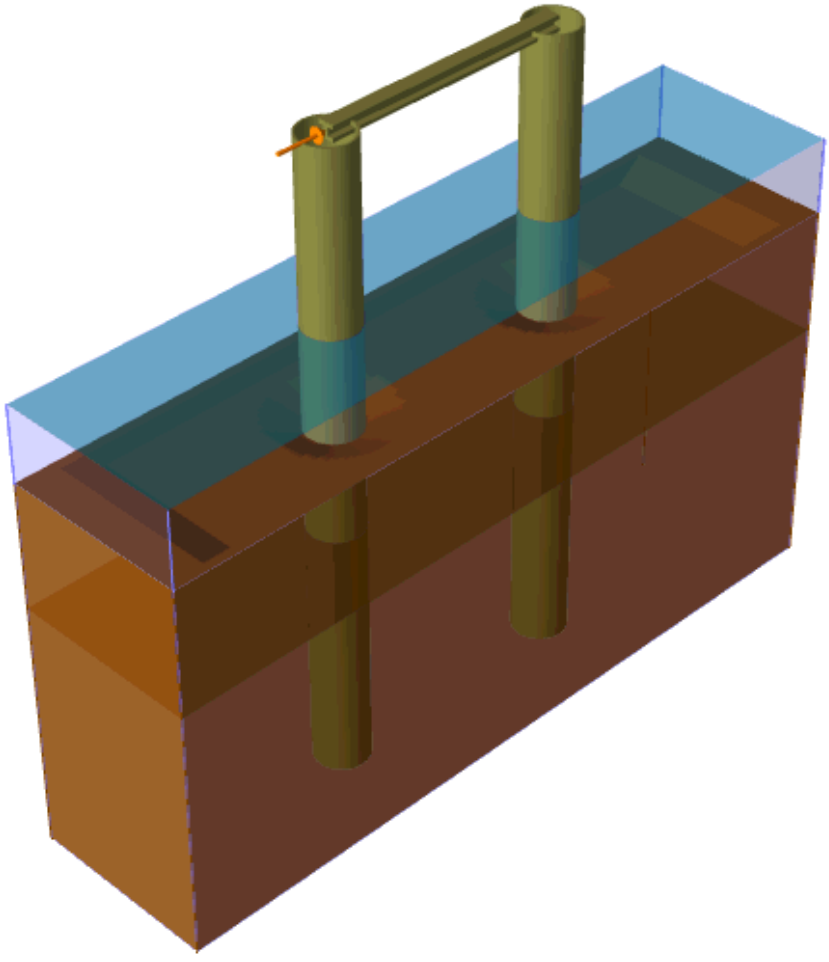
The ‘Z Bottom’ values, i.e. the Z-coordinate of each layer, are expressed with respect to the global coordinate system, or with respect to the seabed layer if the ‘Soil layers relative to seabed Z’ is checked. Inversely, by checking ‘Soil layers relative to seabed Z’ the existing values will automatically be transformed so as to be relative to the seabed Z.

Applying the settings shown in the above figure we get the following js-commands:

```
Location1 = Location(3 m, 0 m);
Location1.gravity = 9.80665 m/s^2;
Location1.air().density = 0.001226 tonne/m^3;
Location1.air().kinematicViscosity = 1.462e-005 m^2/s;
```

```
Location1.water().density = 1.025 tonne/m^3;
Location1.water().kinematicViscosity = 1.19e-006 m^2/s;
Location1.seabed().normaldirection = Vector3d(0 m,0 m,1 m);
Location1.seabed().seabedDelta = Scour1;
Location1.insertSoilBorder(-5 m);
Location1.insertSoilBorder(-15 m);
Location1.soil(1).soilCurves = SoilCurves1;
Location1.soil(1).soilData = SoilData1;
Location1.soil(1).soilType = Sand1;
Location1.soil(1).numberOfSublayers = 3;
Location1.soil(2).soilCurves = SoilCurves2;
Location1.soil(2).soilData = SoilData2;
Location1.soil(2).soilType = Clay1;
Location1.soil(2).numberOfSublayers = 5;
```

The result is shown in the following figures.



the water surface at z=3 m

the scour at z=0 m

the layer

Sand1,SoilCurves1,SoilData1 is

From z=0 m to z=-5 m and has 3 sublayers

the layer

Clay1,SoilCurves2,SoilData2 is

From z=-5 m to z=-15 m and has 5 sublayers

Figure 3.13: The Soil layers with the structure

Figures 3.12 and 3.13 show exactly where the scour is with respect to the seabed and its 3D geometry.

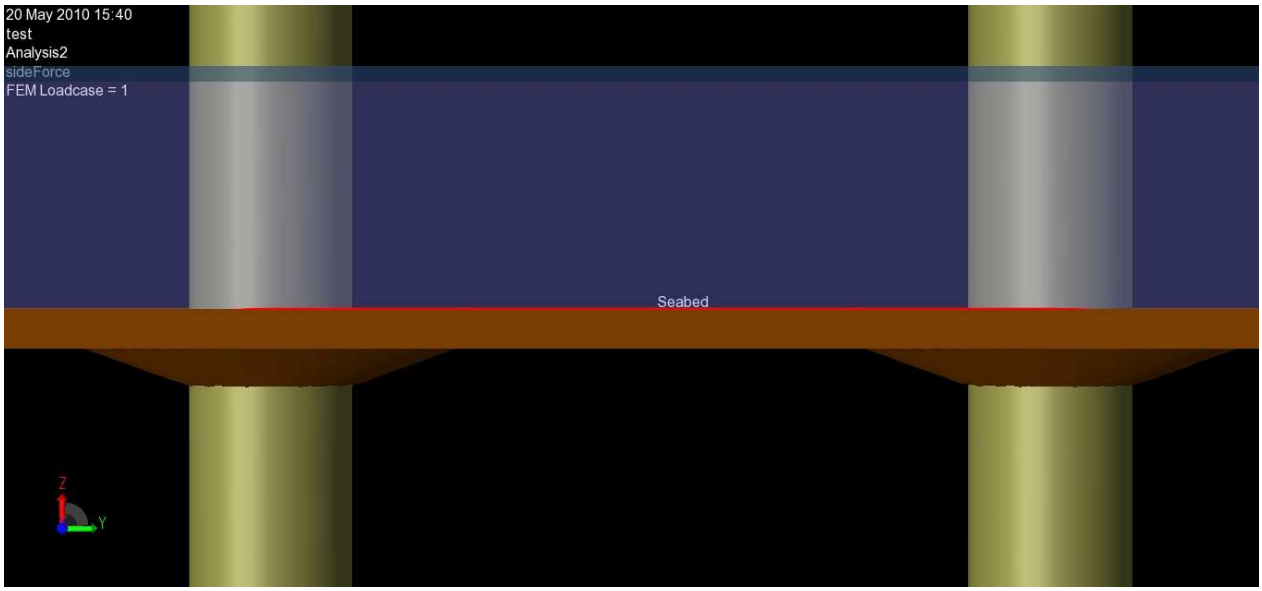


Figure 3.14: The scour in the lateral view (viewed from X)

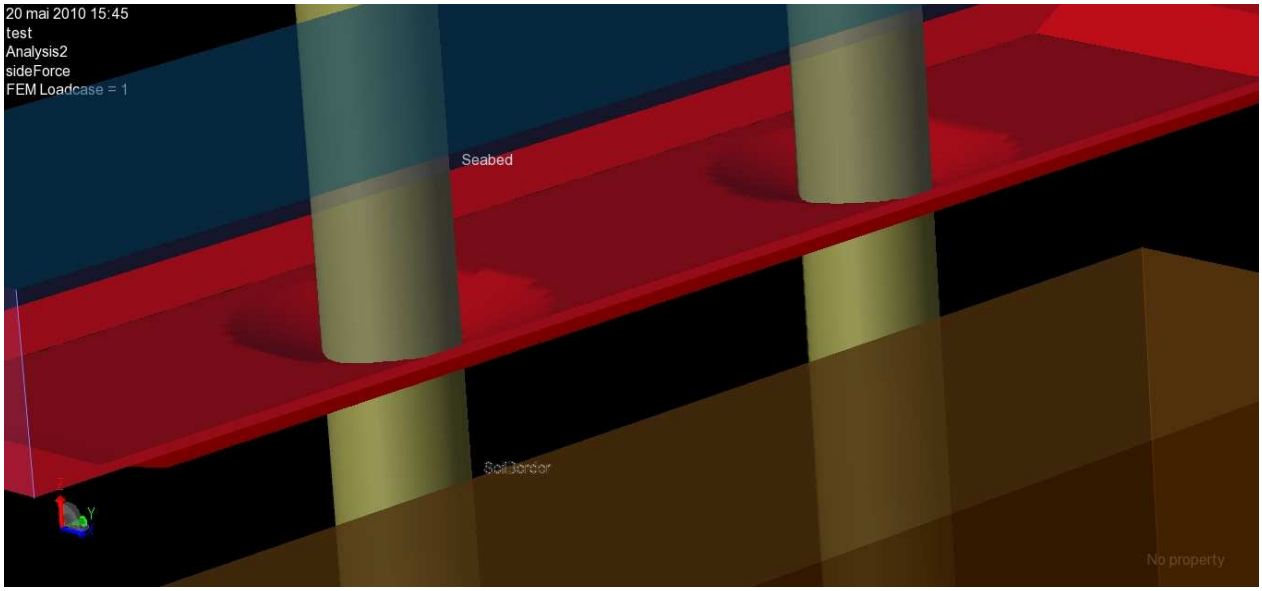


Figure 3.15: The scour in the isometric view

3.6 Create the Condition

A Condition must exist before a Pile Soil Analysis can be created. Via the Condition the Location is identified. In this example a 'CalmSea' Condition is created. Select 'New Wave Load Condition…' to activate the dialog. When the dialog appears you must select 'CalmSea' in the 'Wave model' column and press 'Apply'. No 'Regular wave set' or any other data needs to be given in this case.

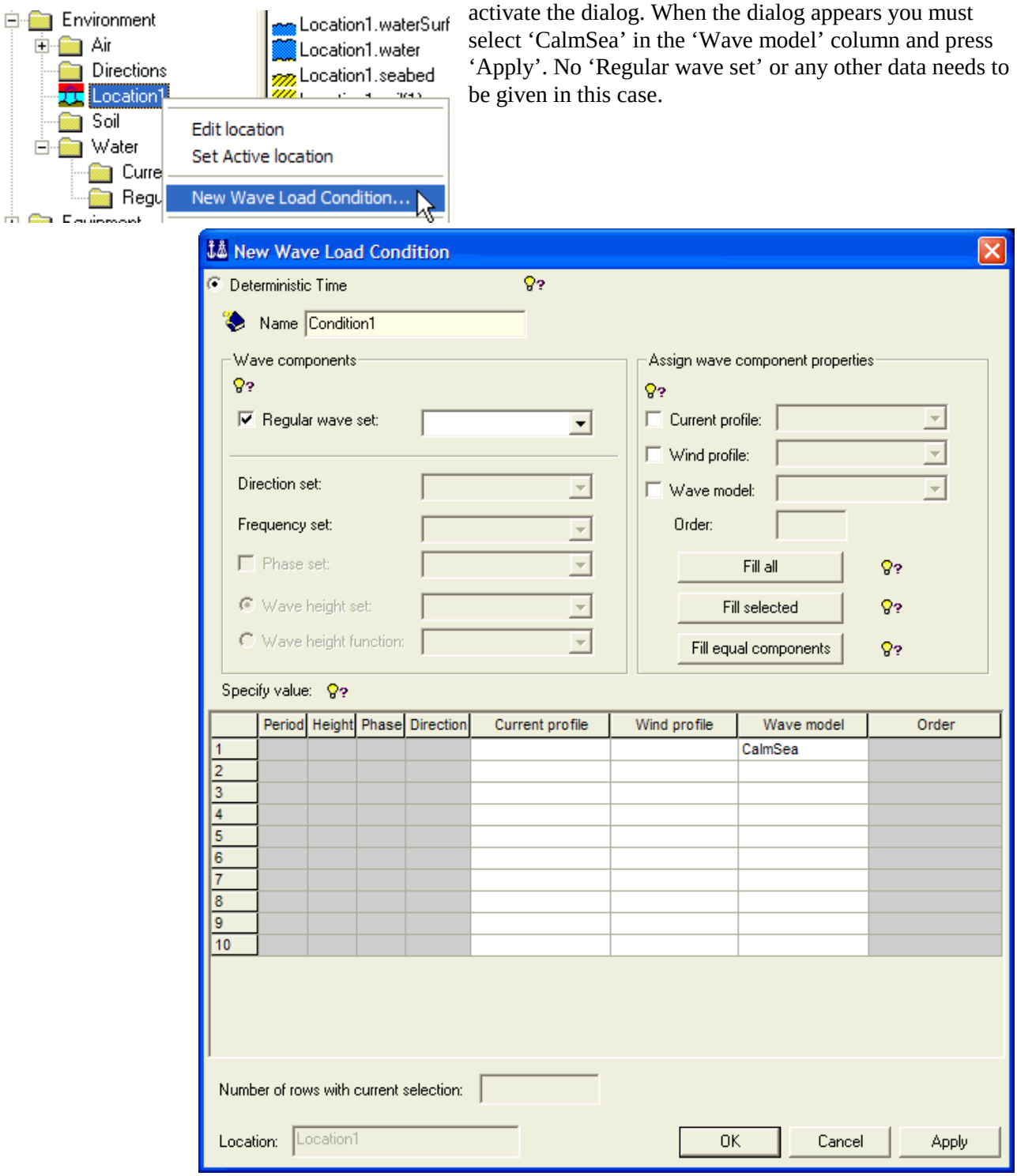


Figure 3.16: Creating the 'CalmSea' Condition

The above settings lead to the following js-commands:

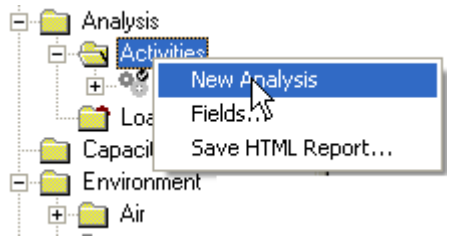
```
Condition1 = DeterministicTime(Location1);
Condition1.addCalmSea();
```

4. PILE SOIL ANALYSIS

This chapter describes how to create and execute the Pile Soil Analysis.

4.1 Create the Pile-Soil Analysis

Select the "New Analysis…" entry on the RMB menu, over the browser node "Activities".



The dialog for creating an Analysis appears and looks as in Figure 4.1 when 'Pile Soil Analysis' is checked. Select 'Wave Load Condition', check the 'Superelement levels' and press 'OK'.

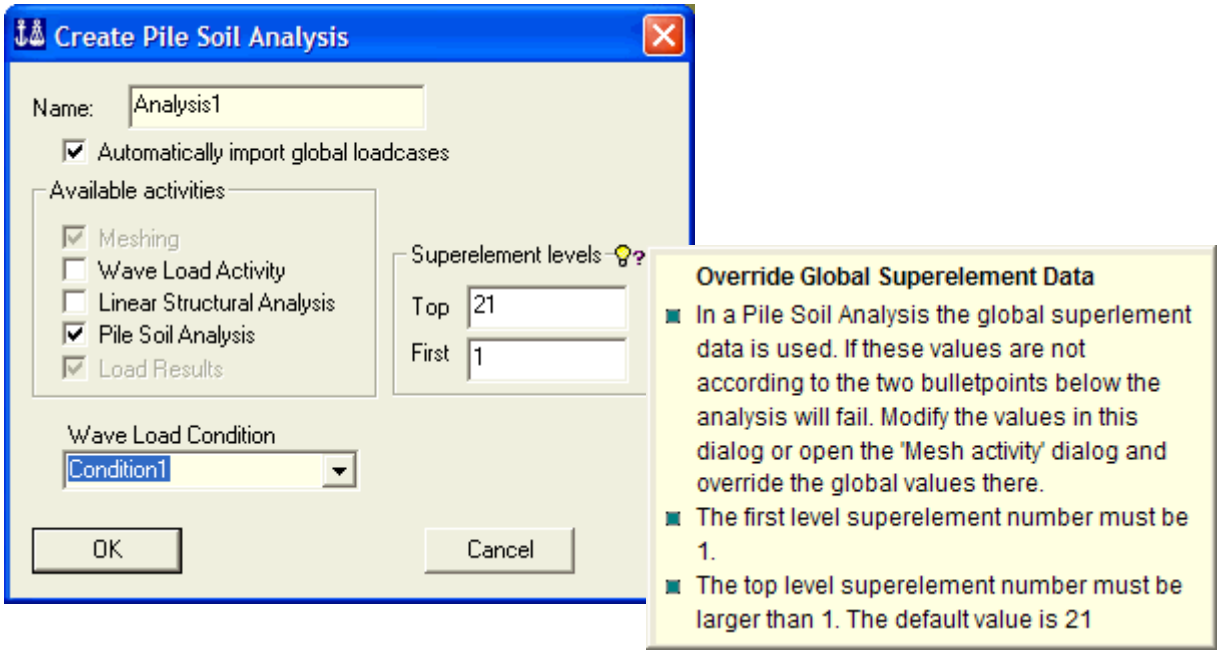

Create Pile Soil Analysis
Name:
Analysis1
Automatically import global loadcases
Available activities
Meshing
Wave Load Activity
Linear Structural Analysis
Pile Soil Analysis
Load Results
Superelement levels
Top
21
First
1
Wave Load Condition
Condition1
OK
Cancel
Override Global Superelement Data
In a Pile Soil Analysis the global superlement data is used. If these values are not according to the two bulletpoints below the analysis will fail. Modify the values in this dialog or open the 'Mesh activity' dialog and override the global values there.
The first level superelement number must be 1.
The top level superelement number must be larger than 1. The default value is 21

Figure 4.1: The dialog for creating the 'Pile Soil Analysis'

Note the following:

- If the superelement numbers are not correct the analysis will not run.

- A wave load condition must be selected. The condition defines the location of the analysis.

The above settings lead to the following js-command:

```
Analysis1 = Analysis(true);
Analysis1.add(MeshActivity());
Analysis1.step(1).superElementTypeTop = 21;
```

```
Analysis1.step(1).superElementType = 1;
Analysis1.add(PileSoilAnalysis(Condition1));
Analysis1.add(LoadResultsActivity());
```

Edit the Pile Soil Analysis

It may be necessary to modify some of the default settings of the Pile Soil Analysis. This is done in the 'Edit Pile Soil Analysis' dialog. Select 'Edit Pile Soil Analysis…' to activate the dialog.

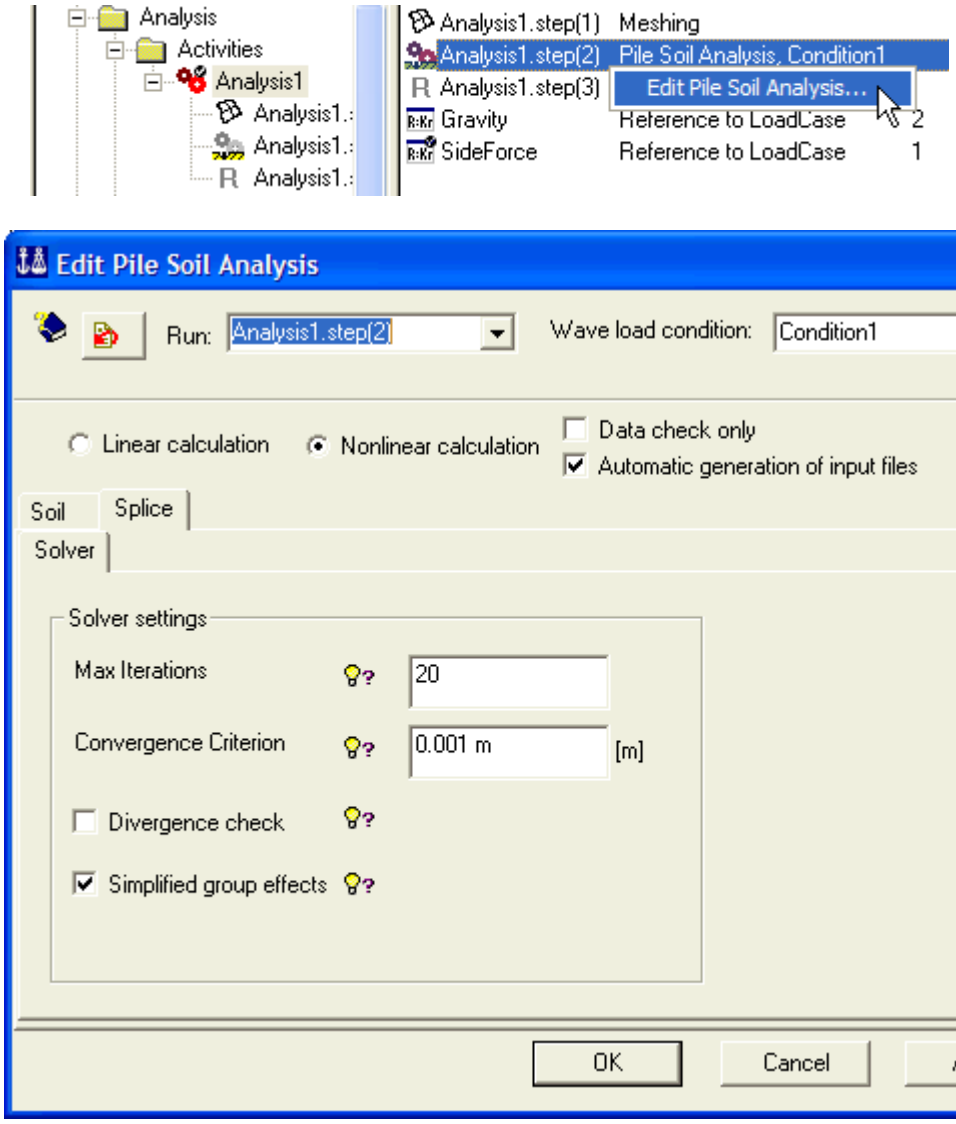


Figure 4.2: The 'Edit Pile Soil Analysis' dialog

The Soil tab of this dialog will not be described in this training example as there are no new features here. However, two new Splice Solver control data options have been introduced, the *Divergence check* and the *Simplified group effects*. Both the new options may stabilize and speed up the calculations.

Divergence check

Option to activate/deactivate divergence check. The checkbox is unchecked by default.

Checkbox unchecked:

- No divergence check is performed.

Checkbox checked:

- Splice automatically checks if convergence is achieved. If after 3 iterations the solution does not improve, the iterations are stopped and splice will terminate for this loadcase.
- This means that Splice returns a solution even if the result does not converge.

Simplified group effects

Option to activate/deactivate simplified calculations of earth sliding due to group effects. Splice is instructed as to how pile/soil/pile interaction calculations shall be performed. The checkbox is checked by default.

Checkbox unchecked:

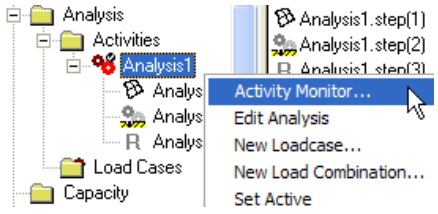
- This is the “normal” procedure. After each iteration the forces transmitted from each pile element to the soil are computed, and used at the start of next iteration to compute new incremental displacements of the soil volume surrounding all pile nodes.

Checkbox checked:

- For this case the system is first solved neglecting pile/soil/pile interaction by a number of iterations. When the convergence criterion has been satisfied, the forces transmitted from each pile element to the soil are computed. At the start of the next iteration the corresponding incremental soil displacements are computed. The iterations are then continued until the convergence criterion is again satisfied, or the maximum no of iterations has been reached, keeping the computed incremental soil displacement unchanged. This approach can be used to solve problems that tend to diverge, or simply to save computer time.

4.2 Executing the Pile Soil Analysis

To execute the analysis select the “Activity Monitor…” entry on the RMB menu, over the relevant ‘Analysis’ browser node. The ‘Activity Monitor’ appears as shown below.



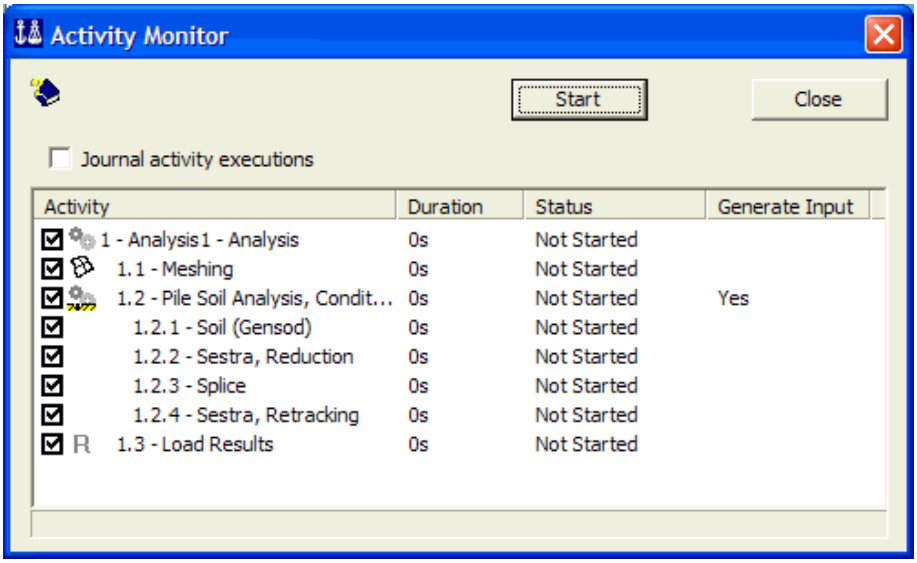

Activity Monitor
Start
Close
Journal activity executions
Activity
Duration
Status
Generate Input
1 - Analysis1 - Analysis 0s Not Started
1.1 - Meshing 0s Not Started
1.2 - Pile Soil Analysis, Condit... 0s Not Started Yes
1.2.1 - Soil (Gensod) 0s Not Started
1.2.2 - Sestra, Reduction 0s Not Started
1.2.3 - Splice 0s Not Started
1.2.4 - Sestra, Retracking 0s Not Started
1.3 - Load Results 0s Not Started

Figure 4.3: The Activity Monitor -before execution

For executing the analysis press ‘Start’.

After execution has finished input to and/or output from many of the activities may be inspected by RMB on the activity. Figure 4.4 shows the RMB menu of the Soil (Gensod) activity.

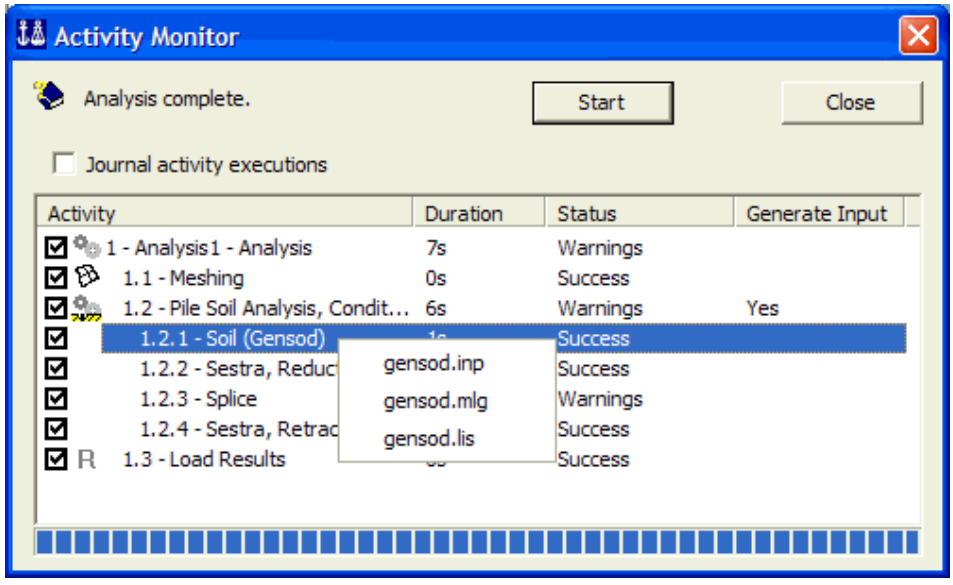


Figure 4.4: After execution - RMB on the Soil activity

5. THE COMMAND INPUT FILE

```
//Exported using: GeniE D5.2-04 started 18-Jun-2010 13:03:25

GenieRules.Tolerances.useTolerantModelling = true;
GenieRules.Compatibility.version = "V5.2-04";
GenieRules.Meshing.autoSimplifyTopology = true;

//Units
GenieRules.Units.setDatabaseUnits("m", "kN", "delC");
GenieRules.Units.setInputUnit(Angle, "deg");
GenieRules.Units.setInputUnit(Force, "kN");
GenieRules.Units.setInputUnit(Length, "m");
GenieRules.Units.setInputUnit(TempDiff, "delC");

//***** PROPERTIES *****//
//Sections
I600 = ISection(0.6 m, 0.6 m, 0.15 m, 0.15 m);
Pipe21 = PipeSection(2.1 m, 0.08 m);

//Materials
Steel = MaterialLinear(255000000 KPa, 7850 tonne/m^3, 2.1e+011 KPa, 0.3, 1.2e-005 delC^-1, 0.03
kN*s/m);

//Pile Characteristics
PileType1 = PileCharacteristics(1000 tonne/m^3, tcInfinite);

//Seabed Deltas
Scour1 = Scour(0.5 m, 1 m, 20 deg);

//Soil Types
Clay1 = Clay(false, 1, 1.94 tonne/m^3, 0.01, 200 KPa, 0 m, 150 KPa, -15 m);
Clay1.apiJFactor = 0.5;

Sand1 = Sand(false, 1, 1.99 tonne/m^3, 20 deg);

//Soil Data and Soil Curves
SoilCurves1 = SoilCurves(pyManual, tzManual, qzManual);
SoilCurves1.addManualPY(2.1 m, Array(0 m, 0.02101 m, 0.04202 m, 0.06303 m, 0.08404 m, 0.105 m,
0.1261 m, 0.1471 m, 0.1681 m, 0.1891 m, 0.3 m), Array(0 KPa, 35.64 KPa, 61.51 KPa, 76.26 KPa, 83.52
KPa, 86.85 KPa, 88.32 KPa, 88.96 KPa, 89.23 KPa, 89.35 KPa, 89.44 KPa));
SoilCurves1.addManualTZ(2.1 m, Array(0 m, 0.005 m, 0.15 m), Array(0 KPa, 15 KPa, 15 KPa));
SoilCurves1.addManualQZ(2.1 m, Array(0 m, 0.0042 m, 0.0273 m, 0.0882 m, 0.1533 m, 0.21 m, 0.3 m),
Array(0 KPa, 5 KPa, 10 KPa, 15 KPa, 18 KPa, 20 KPa, 20 KPa));
SoilCurves2 = SoilCurves(pyManual, tzManual, qzManual);
SoilCurves2.addManualPY(2.1 m, Array(0 m, 0.0021 m, 0.0042 m, 0.0063 m, 0.01418 m, 0.02205 m,
0.02993 m, 0.0378 m, 0.0441 m, 0.1134 m, 0.3 m), Array(0 KPa, 409.9 KPa, 579.8 KPa, 710 KPa, 931.8
KPa, 1011 KPa, 1021 KPa, 985.3 KPa, 917 KPa, 160.8 KPa, 160.8 KPa));
SoilCurves2.addManualTZ(2.1 m, Array(0 m, 0.00336 m, 0.00651 m, 0.01197 m, 0.0168 m, 0.021 m, 0.15
m), Array(0 KPa, 65 KPa, 108.3 KPa, 162.5 KPa, 195 KPa, 216.7 KPa, 216.7 KPa));
SoilCurves2.addManualQZ(2.1 m, Array(0 m, 0.0042 m, 0.0273 m, 0.0882 m, 0.1533 m, 0.21 m, 0.3 m),
Array(0 KPa, 250 KPa, 500 KPa, 750 KPa, 900 KPa, 1000 KPa, 1000 KPa));
```

```
SoilData1 = SoilData(-1 KPa, 0.5, 15 KPa, 11 KPa, 0.01, 20 KPa, 0.05);
SoilData2 = SoilData(-1 KPa, 0.5, 150 KPa, 250 KPa, 125 KPa, 225 KPa, 0 m, -15 m, 0.01, 1000 KPa,
0.05);

//***** RULES *****//
//Connected Move Rules
GenieRules.ConnectedMove.useStructuralPoints = false;
GenieRules.ConnectedMove.defaultConnected = false;
GenieRules.ConnectedMove.rearrangeXJoints = false;

//Geometry Rules
GenieRules.Geometry.beamTopologySnapping = true;
GenieRules.Geometry.guideCurveTopologySnapping = true;
GenieRules.Geometry.creationGrouping = cgGroupingOff;

//Joint Creation Rules
GenieRules.JointCreation.autoGenerate = false;
GenieRules.JointCreation.selectionAware = false;
GenieRules.JointCreation.exclude(geFreeThroughBeams, true);
GenieRules.JointCreation.exclude(geThroughBeamPure, true);
GenieRules.JointCreation.exclude(geThroughBeams, false);
GenieRules.JointCreation.exclude(geFreeBeamEnds, true);
GenieRules.JointCreation.exclude(ge2BeamAligned, true);
GenieRules.JointCreation.exclude(geBeamEnds, false);

//JointDesign Rules
GenieRules.JointDesign.setDefaultCanRule(0.25, 0.3 m);
GenieRules.JointDesign.setDefaultStubRule(1, 0.6 m);
GenieRules.JointDesign.coneAngle = 9.462322208 deg;
GenieRules.JointDesign.minimumGap = 0.0508 m;
GenieRules.JointDesign.gapTolerance = 0.001 m;
GenieRules.JointDesign.planeTolerance = 1 deg;
GenieRules.JointDesign.braceAngleMoveLimit = 10 deg;
GenieRules.JointDesign.chordAlignmentTolerance = 5 deg;
GenieRules.JointDesign.flushBraces = false;
GenieRules.JointDesign.iterations = 2;

//Local Joint Flexibility (LJF) Rules
GenieRules.LJF.method = ljfBuitrago1993;
GenieRules.LJF.setLimit(ljfAxial, 0.1, 5);
GenieRules.LJF.setLimit(ljfIPB, 0.1, 5);
GenieRules.LJF.setLimit(ljfOPB, 0.1, 5);

//Meshing rules
GenieRules.Meshing.elementType = mp1stOrder;
GenieRules.Meshing.superElementType = 1;
GenieRules.Meshing.autoSimplifyTopology = true;
GenieRules.Meshing.autoSplitPeriodicGeometry = false;
GenieRules.Meshing.preference(mpPreferRectangularMesh, false);
GenieRules.Meshing.preference(mpAllowTriangularElements, true);
```

```
GenieRules.Meshing.preference(mpPreferPointMassAsNodeMass, true);
GenieRules.Meshing.preference(mpUseDrillingElements, false);
GenieRules.Meshing.preference(mpUseEccentricHinges, false);
GenieRules.Meshing.eliminateInternalEdges = false;
GenieRules.Meshing.eliminateInternalVertices = true;
GenieRules.Meshing.preference(mpIncludeUnusedProperties, false);
GenieRules.Meshing.preference(mpUseLongLoadcaseNames, false);
GenieRules.Meshing.preference(mpUseLongSetNames, false);
GenieRules.Meshing.preference(mpUseLongPropertyNames, false);
GenieRules.Meshing.preference(mpMeshDensityRounded, false);
GenieRules.Meshing.scantlings = msGross;
GenieRules.Meshing.ignoreEccentricities = false;
GenieRules.Meshing.useCocentricBeams = false;
GenieRules.Meshing.faceMeshStrategy = SesamQuadMesher;
GenieRules.Meshing.edgeMeshStrategy = UniformDistributionEdge;
GenieRules.Meshing.activate(mpMaxAngle, mpFail, true);
GenieRules.Meshing.setLimit(mpMaxAngle, mpFail, 179 deg);
GenieRules.Meshing.activate(mpMaxAngle, mpSplit, false);
GenieRules.Meshing.setLimit(mpMaxAngle, mpSplit, 165 deg);
GenieRules.Meshing.activate(mpMinAngle, mpFail, false);
GenieRules.Meshing.setLimit(mpMinAngle, mpFail, 1 deg);
GenieRules.Meshing.activate(mpMinAngle, mpSplit, false);
GenieRules.Meshing.setLimit(mpMinAngle, mpSplit, 15 deg);
GenieRules.Meshing.activate(mpMaxRelativeJacobi, mpFail, false);
GenieRules.Meshing.setLimit(mpMaxRelativeJacobi, mpFail, 10);
GenieRules.Meshing.activate(mpMaxRelativeJacobi, mpSplit, false);
GenieRules.Meshing.setLimit(mpMaxRelativeJacobi, mpSplit, 5);
GenieRules.Meshing.activate(mpMinNormalizedJacobi, mpFail, false);
GenieRules.Meshing.setLimit(mpMinNormalizedJacobi, mpFail, 0);
GenieRules.Meshing.activate(mpMinNormalizedJacobi, mpSplit, false);
GenieRules.Meshing.setLimit(mpMinNormalizedJacobi, mpSplit, 0.2);
GenieRules.Meshing.activate(mpMinEdge, false);
GenieRules.Meshing.setLimit(mpMinEdge, 0.1);
GenieRules.Meshing.activate(mpMaxChord, false);
GenieRules.Meshing.setLimit(mpMaxChord, 0.2);
GenieRules.Meshing.activate(mpMaxTwistAngle, mpFail, false);
GenieRules.Meshing.setLimit(mpMaxTwistAngle, mpFail, 30 deg);
GenieRules.Meshing.activate(mpMaxTwistAngle, mpSplit, false);
GenieRules.Meshing.setLimit(mpMaxTwistAngle, mpSplit, 10 deg);
GenieRules.Meshing.basicLCfactor = 1;
GenieRules.Meshing.analysisFolders = true;
GenieRules.Meshing.preference(mpAdjustNumberOfElements, true);

//Tolerances Rules
GenieRules.Tolerances.angleTolerance = 0 deg;
GenieRules.Tolerances.pointTolerance = 0.01 m;
GenieRules.Tolerances.useTolerantModelling = true;

//Set Rules
GenieRules.Sets.scriptCompact = true;
```

```
//***** STRUCTURE *****//
//Beams
Steel.setDefault();
I600.setDefault();
Bm1 = Beam(Point(0 m,0 m,10 m), Point(0 m,10 m,10 m));

Pipe21.setDefault();
Bm2 = Beam(Point(0 m,0 m,10 m), Point(0 m,0 m,0 m));
Bm3 = Beam(Point(0 m,10 m,10 m), Point(0 m,10 m,0 m));
Pile1 = Pile(Point(0 m,0 m,0 m), Point(0 m,0 m,-15 m));
Pile1.pileCharacteristics = PileType1;
Pile2 = Pile(Point(0 m,10 m,0 m), Point(0 m,10 m,-15 m));
Pile2.pileCharacteristics = PileType1;

//Joints
Jt1 = Joint(Point(0 m,0 m,10 m));

//***** GUIDING GEOMETRY *****//
//***** ENVIRONMENT *****//
//Locations
Location1 = Location(3 m, 0 m);
Location1.gravity = 9.80665 m/s^2;
Location1.air().density = 0.001226 tonne/m^3;
Location1.air().kinematicViscosity = 1.462e-005 m^2/s;
Location1.water().density = 1.025 tonne/m^3;
Location1.water().kinematicViscosity = 1.19e-006 m^2/s;
Location1.seabed().normaldirection = Vector3d(0 m,0 m,1 m);
Location1.seabed().seabedDelta = Scour1;
Location1.insertSoilBorder(-5 m);
Location1.insertSoilBorder(-15 m);
Location1.soil(1).soilCurves = SoilCurves1;
Location1.soil(1).soilData = SoilData1;
Location1.soil(1).soilType = Sand1;
Location1.soil(1).numberOfSublayers = 3;
Location1.soil(2).soilCurves = SoilCurves2;
Location1.soil(2).soilData = SoilData2;
Location1.soil(2).soilType = Clay1;
Location1.soil(2).numberOfSublayers = 5;

//Conditions
Condition1 = DeterministicTime(Location1);
Condition1.water().setNoCurrent();
Condition1.populate();
Condition1.addCalmSea();
Condition1.component(1).water().setNoCurrent();

//***** EQUIPMENTS *****//
//***** SETS ( Create ) *****//
//***** LOAD MODELLING AND ANALYSIS *****//
SideForce = LoadCase();
```

```
SideForce.setFemLoadcase(1);
Gravity = LoadCase();
Gravity.setFemLoadcase(2);
SideForce.excludeSelfWeight();
SideForce.includeStructureMassWithRotationField();
//  Loads
PLoad1 = PointLoad(SideForce, Point(0 m,0 m,10 m), 0 kN, 100 kN, 0 kN, 0 kN*m, 0 kN*m, 0 kN*m);

Gravity.includeSelfWeight();
Gravity.includeStructureMassWithRotationField();

//Analyses
Analysis1 = Analysis(true);
Analysis1.add(MeshActivity());
Analysis1.step(1).beamsAsMembers = true;
Analysis1.step(1).smartLoadCombinations = true;
Analysis1.step(1).pileBoundaryCondition = pmPileSoilInteraction;
Analysis1.step(1).superElementTypeTop = 21;
Analysis1.step(1).superElementType = 1;
Analysis1.add(PileSoilAnalysis(Condition1));
Analysis1.step(2).soil.materialCoeffs.tanPhiCoeff = 1;
Analysis1.step(2).soil.materialCoeffs.shearStrengthCoeff = 1;
Analysis1.step(2).soil.materialCoeffs.skinFrictionCoeff = 1;
Analysis1.step(2).soil.materialCoeffs.pileTipResistanceCoeff = 1;
Analysis1.step(2).soil.curveGeneration.lowestShearStiff = 100 KPa;
Analysis1.step(2).soil.curveGeneration.lowestLevelWithCyclicPY = 0 m;
Analysis1.step(2).soil.curveGeneration.zoneOfInfluenceTZ = 10;
Analysis1.step(2).soil.curveGeneration.curveShapeFactorTZ = 0.9;
Analysis1.step(2).soil.groupEffects.averagePoissonRatio = 0.5;
Analysis1.step(2).soil.groupEffects.modulusOfElasticity(10000 KPa, 0 m, 10000 KPa, -100 m);
Analysis1.step(2).soil.loadsAtSurface.verticalStressAtSurface = 0 KPa;
Analysis1.step(2).soil.loadsAtSurface.verticalStressUnderEmbankment = 0 KPa;
Analysis1.step(2).soil.loadsAtSurface.widthOfEmbankmentSlopingPart = 0 m;
Analysis1.step(2).soil.loadsAtSurface.distancePileToEmbankmentToe = 0 m;
Analysis1.step(2).soil.loadsAtSurface.verticalStressUnderCircularLoadedArea = 0 KPa;
Analysis1.step(2).soil.loadsAtSurface.radiusOfCircularLoadedArea = 0 m;
Analysis1.step(2).splice.solver.maxIterations = 20;
Analysis1.step(2).splice.solver.convergenceCriterion = 0.001 m;
Analysis1.step(2).splice.solver.divergenceCheck = false;
Analysis1.step(2).splice.solver.simplifiedGroupEffects = true;
Analysis1.add(LoadResultsActivity());


//***** LOAD INTERFACES *****//
//***** MODEL VIEWS *****//
//***** SETS ( Fill ) *****//
```